Long Form Question Answering Dataset Creation for Business Use Cases using Noise-Added Siamese-BERT

Tolga Çekiç[1,*], Yusufcan Manav[1,*], Batu Helvacıoğlu[1], Enes Burak Dündar[1], Onur Deniz[1] and Gülşen Eryiğit[2]

[1]*Yapı Kredi Teknoloji, Istanbul, Turkey*

[2]*Istanbul Technical University, Department of AI&Data Eng., Istanbul, Turkey*

Keywords: Long Form Question Answering, Siamese-BERT, Dense Passage Retrieval.

Abstract: In business cases, there is an increasing need for automated long form question answering (LFQA) systems from business documents, however data for training such systems is not easily achievable. Developing such data sets require a costly human annotation stage where <<question-answer-related document passage>> triplets should be created. In this paper, we present a method to rapidly develop an LFQA dataset from existing logs of help-desk data without need of manual human annotation stage. This method first creates a Siamese-Bert encoder to relate recorded answers with business documents' passages. For this purpose, the Siamese-Bert encoder is trained over a synthetically created dataset imitating paraphrased document passages using a noise model. The encoder is then used to create the necessary triplets for LFQA from business documents. We train a Dense Passage Retrieval (DPR) system using a bi-encoder architecture for the retrieval stage and a cross-encoder for re-ranking the retrieved document passages. The results show that the proposed method is successful at rapidly developing LFQA systems for business use cases, yielding a 85% recall of the correct answer at the top 1 of the returned results.

1 INTRODUCTION

Automated question answering (QA) systems have been in demand for a long time. They can be crucial in use cases ranging from search engines to conversational systems. Recent advances in natural language processing and machine learning made it possible to create successful QA systems. Most QA systems (i.e., factoid QA) are developed for questions requiring short-form factual answers , namely answers usually consisting of a word or a phrase, e.g. *Q: What is the capital of France? A: Paris.* However, especially in domain-specific areas, questions may require an answer that contains detailed information such as the explanation of a certain procedure. Task of answering such questions that require a detailed answer is named as long-form question answering (LFQA). Recent studies have been started to work in this emergent area. However creating datasets for LFQA tasks can be quite costly and time consuming. The only publicly available large-scale dataset for LFQA so far is the "Explain Like I'm Five" (ELI5) dataset (Fan et al., 2019) which is derived from Reddit questions.

The need for LFQA rises rapidly, especially in business use cases. In business environments, large amounts of QA data are naturally generated and logged daily. These QA data may consist of questions requiring factoid answers (e.g., customer questions related to their transactions) or long-form answers. Factoid questions are generally performed by querying structured databases, whereas long-form questions need most of the time to be answered from unstructured business documents. In case of LFQA, questions themselves may be also long in form describing a specific case (e.g., the procedure to be taken in case of a credit application) in addition to the required answers. Training of LFQA systems (Karpukhin et al., 2020; Guu et al., 2020) requires annotation of relevant passages within supporting documents for generating answers, which is missing in these existing business QA datasets. However, it could be possible to automatically develop these annotations by using NLP techniques and train successful LFQA systems on these.

[*]Equal Contribution

Çekiç, T., Manav, Y., Helvacıoğlu, B., Dündar, E., Deniz, O. and Eryiğit, G.
Long Form Question Answering Dataset Creation for Business Use Cases using Noise-Added Siamese-BERT.
DOI: 10.5220/0011550900003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 1: KDIR*, pages 75-82
ISBN: 978-989-758-614-9; ISSN: 2184-3228

Differing from open-domain use cases, in business environments (e.g., banking, law), there exist documents (e.g., documentation of bylaws such as process documents, compliance, and regulatory documents, company policies, legal documents such as court decisions), which are frequently used as supporting references by experts while answering questions. Although the information required to answer most questions originate from these documents, only a subset of existing answers contains direct references to these documents. But, these are very valuable to automatically construct training data for LFQA systems.

This paper introduces an approach for rapid development of LFQA datasets for business use cases. It experiments with two similarity methods (BM25 and Siamese-BERT trained on noise-added document passages) to automatically relate the document passages to existing human answers. It then reports the results of an LFQA system using a deep learning document passage retrieval (DPR) architecture trained on the developed dataset.

The paper is structured as follows. In section 2 we present the related work on question answer systems. In section 3, we explain our methodology of creating a dataset for LFQA. In section 4, we present our experimental setup and discuss evaluation results. Finally, in section 5 we conclude the paper.

2 RELATED WORK

The question answering literature can be split in to two as Short Form Question Answering and Long Form Question Answering, based on the length of the answer. In the recent years with the help of the transformer based network the performance on the Question Answering problem have been increased rapidly. However most of the works and datasets in this area focused on Short form answers.

Short Form Question Answering(SFQA): The answers in this format are short and generally extracted directly from the source document. Which is easier to evaluate, it can be checked by whether the found answer matches the gold answer.

The methods proposed in SFQA problem generally consist of two parts. First one is a retriever which selects the possible documents that contain the answer, the second part is a machine reading comprehension module which checks the retrieved passages for answering the given question. One such method DPR (Karpukhin et al., 2020) first trains a bi-encoder network using document passages and question for retrieval. Then uses an dense index to retrieve possible passages to the given questions. Then it uses a reader module to extract the short answers from the passages. The Retro-Reader (Zhang et al., 2020), proposes a two step reader module first being Sketchy reader which reads the passage and decides whether it contains answer to the text, then second Intensive Reader to extract the answer span if exists.The RAG (Lewis et al., 2021) expands on DPR via a text generator module which takes question and retrieved documents to generate answers using the retrieved knowledge.

There are multiple datasets for the SFQA task, with varying creation method and structures. In SQUAD (Rajpurkar et al., 2016; Rajpurkar et al., 2018) the dataset created by writing question which are answerable by information in the given passages. Annotators first read the passage then write questions and select the answer span from the passages. The answers are generally very short in the SQUAD. In the TriviaQA (Joshi et al., 2017), questions and answer pairs are created by humans annotators. Then supporting evidence documents are collected using the QA pair, so the questions not written over the given passages. Answer are also longer up to a sentence.

The Natural Questions (Kwiatkowski et al., 2019) uses queries to Google search engine. They first select queries longer than 8 words and sample the ones similar to natural questions by some rules. Then they run the questions over Google search and keep the ones that have Wikipedia page in the top 5 search result. Then humans annotate the question document pairs in 3 steps, first identify good question, then find passage that contains answer and then annotate short answer within this passage.

There are also other datasets on the SFQA task which are WikiQA (Yang et al., 2015), HotpotQA (Yang et al., 2018) and BeerQA (Qi et al., 2021).

Long Form Question Answering(LFQA): The length of the answers are around couple of sentences to a paragraph in this format. The answers can't be directly checked by exact match with ground truth because of the length and complexity. Generally BLEU or Rouge metrics used for evaluation. The LFQA task generally consist of two steps first step being a retrieval step, in which the documents related to the question are retrieved. Then abstractive or extractive summarization methods are utilized to generate paragraph length answers using those supporting documents.

The methods generally focused on the retrieval part of this problem. The Realm (Guu et al., 2020) proposes a retrieval augmented pretraining task for BERT language model to increase retrieval performance. The DPR (Karpukhin et al., 2020) method is also utilized for the retrieval phase of the LFQA task. The combined BART (Lewis et al., 2020) and DPR model also utilized for this task in which the DPR retrievals the related passages then BART model generates the answers.

Contrary to the SFQA the datasets are very scarce in this problem. The only dataset can be found is the ELI5 (Fan et al., 2019). It is generated using the ELI5 (Explain Like I'm 5) subreddit, and for each question in this subreddit they retrieve the answers which are paragraph long. To generate related documents, they use a TFIDF retriever on the CommonCrawl web dump, to retrieve related documents and combine all documents related parts to generate one supporting document.In the KILT (Petroni et al., 2021) benchmark, for the test and development set of the ELI5 they made humans to evaluate those supporting documents based on if they contain enough information to give answer.

3 METHODOLOGY

As explained in the previous section, commonly used public QA datasets like Natural Questions (NQ) and SQUAD are annotated manually: the document passages that contain the correct answers are used by humans to create question-answer-document passage (Q-A-Dp) triples. The need to feed document passages into QA systems stems from the goal of answering previously unseen questions. In addition to this, the answers may be outdated in business use cases, and one should refer to up-to-date business documents to extract valid answers. Although Q-A pairs and supporting document links are accessible in business use cases via historical data such as help desk or customer support conversations, annotating the relevant document passages is a costly effort. This section introduces our proposed method to use these existing data without extra human supervision to rapidly generate a Q-A-Dp triples.

3.1 Data Preparation

As the first step, the entire business documents in the document pool are split into passages; the passage length is chosen as 100 words approximately considering the sentence boundaries dynamically, with 50 words stride so that passages are overlapping. This increases the total passage count but since the splitting process is unsupervised, the chance of splitting a potential answer to different passages is minimized by this approach.

As part of our data preparation step, we filter out the Q-A pairs and select only the ones containing supporting document links and at least 10-words-length text in their answers.

3.2 Dataset Creation

In order to create Q-A-Dp triplets, we utilize the similarity between the answers of the Q-A pairs and document passages. The rationale behind this is that if a supporting document link is provided in an answer, the rest of the answer text would contain either a direct copy or a paraphrase of a segment of the referred document. We use two methods to match the related passages from documents. First one is a basic BM25 method. The second one is a Siamese-BERT approach using a noise model to simulate paraphrasing of the passages. For both methods, first, similar passages to the answer text are retrieved from the entire document pool. Then the top-ranked document passage Dp is related to the Q-A pair if the document containing this Dp is the same as the supporting document link in answer A. The following subsections introduce the details of these two methods.

3.2.1 BM25 Method

Relying on the idea that there could be a considerable token overlap between answers and related document passages, we use BM25 to search over the document passages. Initially, the document passages are indexed using Apache Lucene. The queries are formed with the text of the answers (excluding document URLs, links); the queries are lemmatized to make the system more robust to morphological variations and filtered for stop words.

3.2.2 Noise Added Siamese-BERT Model

Since BM25 directly relies on token matching, it may be inefficient on paraphrased answers using semantically similar but different words. In this method, we use a Siamese-BERT approach using triplet loss (Reimers and Gurevych, 2019) to capture a more contextual and semantic representation of the answers and the passages in order to retrieve more relevant document passages. The reason for choosing a Siamese architecture is because it can be trained to discriminate passages passages based on their meanings regardless of common words used in them.

In order to train a Siamese-BERT encoder using triplet loss, it is needed to create anchor-positive-negative sample triplets. At this stage, we describe a noise model in order to synthetically create different anchor samples from original document passages (Figure 1).

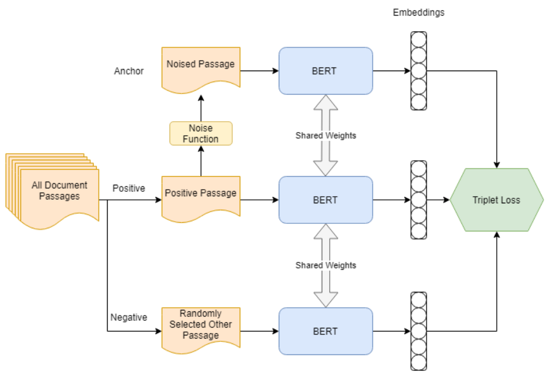

Figure 1: Training of the noise added Siamese-BERT Encoder.

```
Data: passage
Result: noised_passage
sentences=split_to_sentences(passage);
sentences=sentence_dropout(sentences,$\lambda1$);
sentences=sentence_swap(sentences,$\lambda2$);
for sentence in sentences do
    words=split_to_words(sentence);
    words=word_dropout(words,$\lambda3$);
    words=word_replacement_w2v(words,$\lambda4$);

    words=word_replacement_lemma(words,$\lambda5$);

    words=char_level_noise(words,$\lambda6$);
    sentence=combine_sentence(words)
end
noised_passage=generate_passage(sentences)
```

Algorithm 1: Text noising algorithm. $\lambda1$ denotes the sentence dropout probability, $\lambda2$ denotes the random sentence swap probability, $\lambda3$ denotes the probability of the word dropout, $\lambda4$ denotes the probability of replacing a word with a word2vec-closest neighbor, $\lambda5$ denotes the probability of replacing a word with its lemma form, and $\lambda6$ denotes the probability of generating a random character noise within a word.

While the original passages become the positive samples of these anchors, 3 negative samples for each anchor are randomly selected from remaining passages.

Algorithm 1 gives the pseudo-code of our noising function which modifies the input passages according to predefined probabilities for each action. For each passage, the noise model first takes its sentences and randomly eliminates some of them to simulate copied parts of the passages. As the next step, in order to simulate paraphrased sentences, it adds word-level noises by dropping-out some of the words and replacing some of them with the closest neighboring words from a word2vec model trained on our QA pairs and documents or with the lemma of original word. As the final stage, the noise model introduces character level noises to simulate typos. Those are character deletion, addition or replacement of a character with a different one. Each passage obtained from the original documents are noised with this algorithm 10 times to produce 10 randomly generated noised-original passage pairs (i.e., anchor-positive sample pairs).

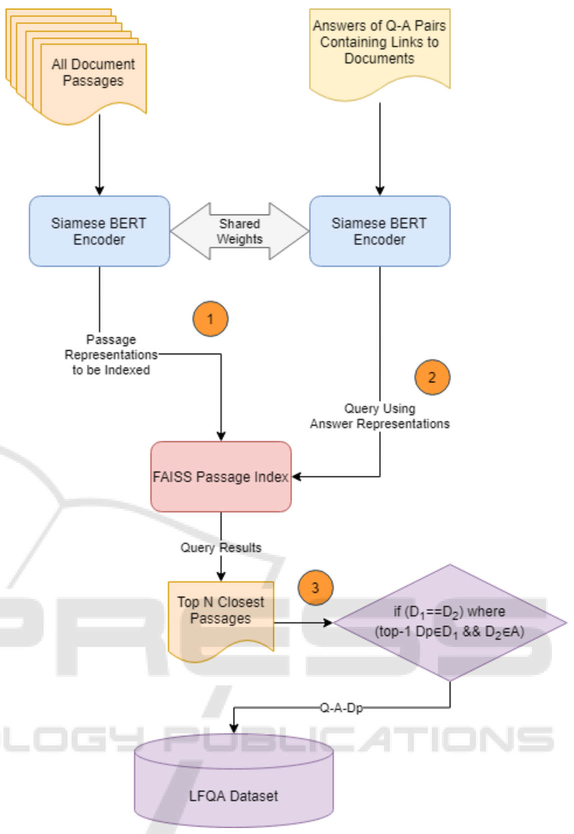
All Document Passages
Answers of Q-A Pairs Containing Links to Documents
Siamese BERT Encoder
Shared Weights
Siamese BERT Encoder
1
Passage Representations to be Indexed
2
Query Using Answer Representations
FAISS Passage Index
Query Results
Top N Closest Passages
3
if (D1==D2) where (top-1 Dp∈D1 && D2∈A)
Q-A-Dp
LFQA Dataset

Figure 2: Flow of the LFQA training dataset creation methodology.
D_1: the document containing the Dp, D_2: the supporting document referred within the answer A.

After training the encoder from the entire passages within our document pool, the original passages are passed from this encoder and indexed using Faiss (Johnson et al., 2017) with their representations. Figure 2 presents the flow of our LFQA dataset creation methodology. Once the document passages are indexed, the answers from original Q-A pairs containing supporting document links are first encoded using the same Siamese-BERT we trained and then their representations are queried over the created Dp index. As explained before, the top-ranked document passage Dp is related to the Q-A pair if the document (D_1 in Figure 2) containing this Dp is the same as the supporting document (D_2 in Figure 2) link in answer A. Finally, the created Q-A-Dp triplets are added to the LFQA dataset.

4 EXPERIMENTAL SETUP

This section introduces the evaluation of the introduced LFQA dataset creation techniques by using them within a deep learning question answering architecture.

4.1 Dataset

Our supporting document pool contains 21K documents. The documents in our document pool are parsed and split to the document passages as described in 3.1. After this splitting operation, a total of 85K passages are obtained. Initial weights for training are obtained from a pretrained BERT model during the training phase of the Siamese-BERT encoder. Similar to (Reimers and Gurevych, 2019), we use the average of word representations to construct passage representations. The noise function produces a training dataset of 850K anchor-positive pairs for training the Siamese-BERT encoder. After the negative sampling, a total of 2.5M anchor-positive-negative triplets are used to train this network. Siamese-BERT encoder is trained for 5 epochs.

As explained in Section 3, we used 2 data sources (Q-A pairs and the document pool) to construct our LFQA dataset, and we used the answers with supporting document links in order to combine them. In our data, the Q-A pairs containing supporting document links (being 71K in total) cover about 5% of the available Q-A pairs. Then we apply our methods in Section 3.2 to extract Question-Answer-Document Passage triplets.

Table 1: Generated triplet counts with different dataset creation models.

Dataset Creation Models	**Generated Q-A-Dp Triplet Count**
BM25	42589
Base-BERT	7495
High Noise Siamese-BERT	15354
Low Noise Siamese-BERT	17722
Low Noise Siamese-BERT (If related Dp is retrieved from Top-3 query results)	23987

Table 1 shows the number of the generated Q-A-Dp triplets (i.e., the size of the generated LFQA dataset) after utilizing the introduced dataset creation models with different parameters. As a baseline method (i.e., Base-BERT), we also test with a pretrained BERT encoder instead of the stated Siamese-BERT encoder (Figure 2). We again use the average of word representations to construct passage representations[1].

We test with different λ probabilities in our noise function (Algorithm 1). We call the model with $\lambda 1 = 0.5, \lambda 2 = 0.4, \lambda 3 = 0.025, \lambda 4 = 0.15, \lambda 5 = 0.05, \lambda 6 = 0.025$ as the High Noise Siamese-BERT. We call the model with $\lambda 1 = 0.1, \lambda 2 = 0.1, \lambda 3 = 0.004, \lambda 4 = 0.012, \lambda 5 = 0.008, \lambda 6 = 0.004$ as the Low Noise Siamese-BERT. As a final model, instead of creating the Q-A-Dp triplets from only the top-1 query result (Figure 2), we test with top-3 results; i.e., if the documents containing the top-3-ranked document passage Dp returned from index query are compared with the supporting document link in the answer, and if one of them is the same with this link, the Dp is included to the Q-A-Dp triplet. As a result, we see that different number of triplets were generated. At this stage, the highest number of mapped answers to the related document passages are obtained with BM25 with 43K retrieved triplets. Since the data we used is classified as confidential we are unable to share the dataset we created.

4.2 Question Answering Architecture

Our next stage is to evaluate the created dataset in a real LFQA scenario to determine if the dataset we created can be used to train a successful model. Our question answering architecture is built on top of a biencoder network which is highly popular on training passage retrieval models (DPR). As explained in Section 2, DPR uses two stages for solving the SFQA problem: a retrieval stage followed by a reader stage. While adapting this model to LFQA, we use only the first stage and aim to extract the relevant document passages. We also chose this model because the size of datasets we created with our method is similar to the size of datasets used in experimentations of this model. In this biencoder architecture, document passages and questions are embedded separately (passage encoder and question encoder) and the networks learns to minimize the similarity distance between positive sample pairs. We trained our network with 7 negative samples for each positive sample, with a batch size of 16 for 100 epochs. After the training, we generate representations of each passage using the trained passage encoder, then index them using Faiss again. At inference time, each question is embedded using the question encoder, and then searched over the established index using dot product similar-

[1]It is also possible to use BERT cls tokens to construct sequence embedding. But since in (Reimers and Gurevych, 2019), using the average of token embeddings are shown to perform better, we preferred this approach as our baseline.

Table 2: Recall at N and human evaluation scores of DPR models trained using our dataset creation models.

Dataset Creation Model	Recall at N						Human Evaluation	
	Top 1	Top 3	Top 5	Top 10	Top 20	Top 100	Top 1	Top 3
BM25	10%	21.8%	28.3%	39.1%	49.9%	73.4%	45%	57%
Base-BERT Model	8.3%	14.9%	18.4%	22.6%	28.4%	45.8%	36%	41%
High Noise Siamese-BERT	16%	28.4%	33.5%	42.3%	50.4%	68.6%	43	55
Low Noise Siamese-BERT	**45.8**%	**63.2**%	**69.7**%	**75.7**%	**81**%	**89.6**%	**68**%	**80**%
Low Noise Siamese-BERT (If related doc passage retrieved in Top 3)	24.3%	37.2%	42.8%	50.2%	58.3%	73.4%	58%	76%
(Low Noise Siamese-BERT re-ranked with Bert cross-encoder)	61%	83.6%	87.1%	88.4%	89%	89.6%	**85**%	**93**%

ity. We retrieve the top 100 passages for our evaluations provided below. As the final stage, we also train a BERT cross-encoder and use it for reranking the top 100 retrieved passages. Cross-encoders are reported to achieve high performances on textual similarity as compared to bi-encoders. However, they are preferred mostly on the reranking stage instead of the retrieval stage due to their low efficiency on high volumes of data (Reimers and Gurevych, 2019).

4.3 Evaluation Method

For the evaluation of these Question Answering models, we first used Top 100 retrieval recall scores of the trained DPR models. After that, since we had limited resources for human testing, we randomly selected 100 examples from the test set and gave human annotators for evaluation. Annotators were given questions and Top 3 retrieval results and evaluated whether those document passage contain the answer to the given question or not.

5 RESULTS & DISCUSSIONS

In the LFQA scenario, the gold answers are not available at inference time differing from the data creation stage. Thus, the document passages should be directly retrieved using only the question text instead of the answers. Table 2 summarizes DPR based QA system performances which are trained with the data synthetically produced with our different dataset creation models. Although BM25 seems to be very successful at mapping the answers to the related documents and their passages by extracting 43K triplets (Table 1), it performs poorly when it comes to the question-answering task, suggesting most of the mapped question-answer pairs are not ideal. We deduce that, since the answers contain very similar keywords to the related document passages, BM25 based method easily maps these. However, the questions are not so and most of the time they contain semantically similar content expressed by different words, and BM25 is not capable of capturing this semantic similarity.

Although the BM25 model produced 2.5 times more training samples (Table 1) than our Siamese-BERT models for training the DPR system, we see that their quality for QA remains low compared to these. When we explore the results further, we see that the BM25 model retrieves commonly passages with titles of the documents rather than related parts since it works on a keyword-based approach which is not helpful to train the QA system to provide an answer directly.

Similar to the findings of the previous literature (Reimers and Gurevych, 2019; Gu et al., 2021), we observe that the Base-BERT model performs even worse than the BM25 model. Also we see that using high noise probabilities were not beneficial for the system. The performance degraded rapidly when we increase the noise probabilities, which shows that the noise probabilities should be selected with care and should be kept low. We understand that increasing the noise does not naturally model the paraphrasing and typos we intended to mimic with this method.

As explained in Section 3.2, as a final model, instead of creating the Q-A-Dp triplets from only the top-1 query result, we test with top-3 results. This approach increases the number of created training samples (Table 1). However, we see that this does not help the DPR system to learn better and the scores decrease drastically. This points out that the lower quality triplets hampers the performance of the model, so selecting only triplets from Top 1 seems to be a better approach.

The last line of the Table 2 provides the scores after reranking the outputs of the best performing model (i.e., Low Noise Siamese-Bert) using the BERT cross-encoder. We see that the reranking stage also drastically improves the results up to 61% at Top 1 in automatic evaluation and 85% Top 1 at human evaluation.

6 CONCLUSIONS

This paper presented our method for rapidly creating LFQA datasets from existing data sources in business environments. Our method trained a Siamese-BERT Model with Noise-Added artificial data to retrieve supporting document passages in order to generate <<question-answer-document passage>> triplets to be used as an LFQA training dataset. We proposed a noise function to obtain altered versions of document passages and trained a Siamese-BERT encoder using these altered passages with the original ones. The model then used this encoder to create the triplets by using existing help-desk logs consisting of supporting document links.

We used the proposed method for creating such a dataset in a real-world business setting together with some baseline retrieval methods such as BM25 and base BERT indexing. For evaluating our approaches, we used the created datasets for training a DPR based question answering system. Both automatic and human evaluation results showed that the DPR model trained with the dataset generated by our methodology outperformed others; the proposed Noise-Added Siamese-BERT model was able to generate better quality LFQA results using fewer training data samples.

REFERENCES

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M. (2019). ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Gu, N., Gao, Y., and Hahnloser, R. H. R. (2021). Local citation recommendation with hierarchical-attention text encoder and scibert-based reranking.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M.-W. (2020). REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

Petroni, F., Piktus, A., Fan, A., Lewis, P., Yazdani, M., Cao, N., Thorne, J., Jernite, Y., Karpukhin, V., Maillard, J., et al. (2021). Kilt: a benchmark for knowledge intensive language tasks. In *NAACL-HLT*, pages 2523–2544. Association for Computational Linguistics.

Qi, P., Lee, H., Sido, O. T., and Manning, C. D. (2021). Answering open-domain questions of varying reasoning steps from text. In *Empirical Methods for Natural Language Processing (EMNLP)*.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yang, Y., Yih, W.-t., and Meek, C. (2015). WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages

2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhang, Z., Yang, J., and Zhao, H. (2020). Retrospective reader for machine reading comprehension. *arXiv preprint arXiv:2001.09694*.